CLAIMS

We Claim:

1. A method of producing an encrypted version of a binary asset, said method comprising:

generating a unique identifier for said binary asset, said unique identifier being computed from at least a portion of the contents of said binary asset and uniquely identifying said binary asset;

encrypting said binary asset using said unique identifier as a key, said encrypting resulting in said encrypted version of said binary asset; and

providing said unique identifier to decrypt said encrypted version of said binary asset and to verify the integrity of said decrypted version of said binary asset.

2. A method as recited in claim 1 further comprising:

generating a second unique identifier for said encrypted version of said binary asset, said second unique identifier being computed from at least a portion of said encrypted version of said binary asset and uniquely identifying said encrypted version of said binary asset; and

providing said second unique identifier for the retrieval of said encrypted version of said binary asset, whereby said second unique identifier may be used to locate said encrypted version.

3. A method as recited in claim 2 further comprising:

creating a descriptor file that includes said unique identifier and said second unique identifier;

generating a third file identifier, said third file identifier being computed from at least a portion of said descriptor file and uniquely identifying said descriptor file;

encrypting said descriptor file using said third file identifier as a key, said encrypting producing an encrypted descriptor file; and

generating a fourth file identifier for said encrypted descriptor file, said fourth file identifier being computed from at least a portion of said encrypted descriptor file and uniquely identifying said encrypted descriptor file, whereby said third file identifier and said fourth file identifier may be used to access the contents of said binary asset.

4. A method of uniquely and securely identifying a computer file, said method comprising:

generating a first file identifier for a file, said first file identifier being computed from at least a portion of said file and uniquely identifying said file;

encrypting said file using said first file identifier as a key, said encrypting producing an encrypted file;

generating a second file identifier for said encrypted file, said second file identifier being computed from at least a portion of said encrypted file and uniquely identifying said encrypted file; and

providing said first file identifier and said second file identifier for the retrieval of said file, whereby said second file identifier may be used to locate said encrypted file, and said first file identifier may be used to decrypt said encrypted file to produce said file.

5. A method as recited in claim 4 wherein said steps of generating use a hash function.

6. A method as recited in claim 4 further comprising:

compressing said file in conjunction with said encrypting.

7. A method as recited in claim 4 further comprising:

creating a descriptor file that includes said first file identifier and said second file identifier;

generating a third file identifier, said third file identifier being computed from at least a portion of said descriptor file and uniquely identifying said descriptor file;

encrypting said descriptor file using said third file identifier as a key, said encrypting producing an encrypted descriptor file; and

generating a fourth file identifier for said encrypted descriptor file, said fourth file identifier being computed from at least a portion of said encrypted descriptor file and uniquely identifying said encrypted descriptor file, whereby said third file identifier and said fourth file identifier may be used to access the contents of said file.

8. A method of uniquely and securely identifying a group of binary assets, a binary asset representing digital information, said method comprising:

computing an intrinsic unique identifier (IUI) for each of said binary assets;

encrypting each of said binary assets using the IUI of each asset as its key to produce an encrypted version of each of said binary assets;

computing an IUI of each of said encrypted versions;

creating a file that includes said IUIs of said binary assets and said IUIs of said encrypted versions;

computing a key IUI for said file;

encrypting said file using said key IUI to produce an encrypted file; and

computing a master IUI for said encrypted file, whereby said key IUI and said master IUI uniquely represent said binary assets and may be used to locate said assets.

9. A method as recited in claim 8 wherein said intrinsic unique identifiers are computed from a portion of the asset or file for which they are computed, and uniquely identify the asset or file for which they are computed.

a1

10. A method as recited in claim 9 wherein each IUI is calculated using a hash function.

11. A method as recited in claim 8 further comprising;

compressing each of said binary assets.

12. A method as recited in claim 8 further comprising:

creating a flattened file that includes said IUIs of said encrypted versions of said binary assets and said master IUI; and

computing a user IUI of said flattened file, whereby a user provided with said user IUI may retrieve said flattened file and thereby retrieve said encrypted versions of said binary assets and retrieve said encrypted file.

13. A descriptor file data structure that reliably identifies a plurality of files, said data structure comprising:

a file name for each of said files;

meta data for each file indicating attributes of each file;

a first intrinsic unique identifier (IUI) for each of said files, each IUI being calculated from the contents of its corresponding file and uniquely identifying its corresponding file; and

a second IUI associated with each of said files, each second IUI being calculated from an encrypted version of its associated file, each file being encrypted using its associated first IUI as a key,

wherein said second IUIs may be used to locate said encrypted versions of said files, and said first IUIs may be used to decrypt said encrypted versions to obtain the non-encrypted versions of said files.

14. A descriptor file as recited in claim 13 wherein said descriptor file is encrypted using its own IUI as a key, said IUI of said descriptor file being calculated from the contents of said descriptor file and uniquely identifying said descriptor file.

15. A method of uniquely and securely identifying a group of files, said method comprising:

creating a key file that includes a plurality of cryptographic keys, each key being associated with one of said group of files;

computing a unique identifier for said key file, said key file identifier being calculated from a portion of the contents of said key file;

encrypting said key file using said key file identifier to produce an encrypted key file;

computing a unique identifier for said encrypted key file, said encrypted key file identifier be calculated from a portion of the contents of said encrypted key file;

creating a flattened file that includes said encrypted key file identifier and unique identifiers for encrypted versions of said files, each unique identifier of one of said encrypted files being calculated from the contents of its associated encrypted file, each

encrypted file having been encrypted using its associated key to encrypted the plaintext version of the file; and

computing a user unique identifier for said flattened file, said user unique identifier be calculated from a portion of the contents of said flattened file, whereby a user provided with said user unique identifier may retrieve said flattened file and said encrypted versions of said files, and when provided with said key file identifier said user may decrypt said encrypted files.

16. A method as recited in claim 15 wherein each of said keys is a unique identifier for its associated file and is calculated from a portion of the contents of its associated file.

17. A method as recited in claim 15 wherein said key file includes meta data for each of said files along with its associated key.

18. A method of reliably retrieving a secure file, said method comprising:

receiving an intrinsic unique identifier for an encrypted version of said file;

retrieving said encrypted version of said file using said IUI of said encrypted versions;

receiving an IUI for the non-encrypted version of said file; and

decrypting said encrypted version of said file using said IUI of said non-encrypted version as a key to obtain the non-encrypted version of said file, whereby said IUI of said encrypted version and said IUI of said non-encrypted version provide access to the contents of said file.

19. A method as receited in claim 18 further comprising:

decompressing said encrypting version of said file in conjunction with said decrypting.

20. A method as recited in claim 18 wherein said intrinsic unique identifiers for said encrypted version and said non-encrypted version are respectively calculated from the contents of said encrypted version and said non-encrypted version.

21. A method of obtaining a data file that has been securely stored, said method comprising:

receiving a master identifier that uniquely identifies an encrypted file;

retrieving said encrypted file using said master identifier;

receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted file;

decrypting said encrypted file using said key identifier to obtain said non-encrypted version, said non-encrypted version including a data file identifier that uniquely identifies a data file and an encrypted version of said data file;

retrieving said encrypted version of said data file using said encrypted identifier; and

decrypting said encrypted data file using said data file identifier as a decryption key, whereby said non-encrypted version of said data file is obtained.

22. A method as recited in claim 21 wherein said non-encrypted file includes meta data for said data file and said method further comprises:

building a portion of a directory structure using said meta data.

23. A method as recited in claim 21 wherein said non-encrypted file includes a plurality of data file identifiers that each uniquely identifies a data file, and a plurality of encrypted identifiers that each uniquely identifies an encrypted version of one of said data files, said method further comprising:

retrieving said encrypted versions of said data files using said encrypted identifiers; and

decrypting said encrypted data files using said data file identifiers as decryption keys.

24. A method as recited in claim 21 further comprising:

calculating a new key identifier for said non-encrypted file; and

comparing said new key identifier to said key identifier to authenticate said non-encrypted file.

25. A method as recited in claim 21 further comprising:

calculating a new data file identifier for said data file; and

comparing said new data file identifier to said data file identifier to authenticate said data file.

26. A method of obtaining a data file that has been securely stored, said method comprising:

receiving a user identifier that uniquely identifies a non-encrypted first file, said non-encrypted first file including a unique identifier identifying an encrypted version of said data file and a master identifier that uniquely identifies an encrypted version of a descriptor file;

retrieving said non-encrypted first file using said user identifier;

retrieving said encrypted descriptor file using said master identifier;

retrieving said encrypted data file using said unique identifier for said encrypted version of said data file;

receiving a key identifier that uniquely identifies the non-encrypted version of said encrypted descriptor file;

decrypting said encrypted descriptor file using said key identifier to obtain said non-encrypted version of said descriptor file, said non-encrypted version including a data file identifier that uniquely identifies said data file; and

decrypting said encrypted data file using said data file identifier as a decryption key, whereby said non-encrypted version of said data file is obtained.

27. A method as recited in claim 26 wherein said non-encrypted descriptor file includes meta data for said data file and said method further comprises:

building a portion of a directory structure for said data file using said meta data.

28. A method as recited in claim 26 wherein said non-encrypted first file includes a plurality of encrypted identifiers that each uniquely identifies an encrypted version of one of a plurality of data files, and wherein said descriptor file includes a plurality of a data file identifiers that each uniquely identifies one of said data files, said method further comprising:

retrieving said encrypted versions of said data files using said encrypted identifiers; and

decrypting said encrypted data files using said data file identifiers as decryption keys.

29. A method as recited in claim 26 further comprising:

calculating a new key identifier for said non-encrypted descriptor file; and

comparing said new key identifier to said key identifier to authenticate said non-encrypted descriptor file.

30. A method as recited in claim 26 further comprising:

calculating a new data file identifier for said data file; and

comparing said new data file identifier to said data file identifier to authenticate said data file.